EasyBook - Software Installation Instructions

A booking system for the Västerås Flygmuseum

Version	Type	Date	Name	Description
1.0	First Version	2015-01-20	Stefano Campanella	
1.1	Adding notes	2015-01-23	Stefano Campanella	Added notes about the code repo on GitHub

The *BookEasy* software is a web-based booking system for the Västerås Flygmuseum. It consists of a dynamic website developed with the PHP programming language and over the Yii framework.

Machine Requirements

The software must be deployed on a web server that meets at least the following technical requirements:

- It is **highly** recommended that the server mounts one of the many flavours of GNU/Linux OS. Windows and Mac are also supported, but the software may have some unexpected behaviour on them.
- the server must have the **Apache (2.2 or above)** web server software properly installed.
    - Apache should have the mod_rewrite module enabled
    - the configuration of the website should have the *AllowOverride* directive set to *All* so that the software can customize the necessary apache options for its workings
    - the configuration of the website should also have the *FollowSymLinks* option enabled to allow mod_rewrite to work properly
    - the software doesn't support deployment as an *Alias*, it must be deployed as a full website or *VirtualHost*
    - It is also possible to use other web servers other than Apache (e.g. nginx), but in this case you should also take care of properly converting the *.htaccess* file in the web/ directory of the software; the files provided are only developed for Apache
- The server must have the **PHP5 (5.4 or above)** software enabled and configured to work with Apache.
    - It is not required any particular flavour of php5, other than having at least version 5.4. Any of the available versions (e.g. mod-php5, php5-fpm, etc) are compatible with the software
    - *mcrypt* extension must be enabled
    - *intl* extension must be enabled
    - *pdo* and *pdo-mysql* extensions must be enabled
- The server must have an installation of **MySql server** (version 5 or above. Better if 5.5 or above) configured to work with php.
    - the server must support the InnoDB storage engine
    - It is not required to have the MySql server on the same machine of Apache and Php. It is just required that php can connect to the MySql server in some way.

- You must have enough permissions on the server to be able to define Scheduled Operations, often known as Cron Jobs on unix-like systems

Software Installation

Since the software is developed over the Yii Framework, its installation steps are the same of the framework, at least for the initial parts:

- Unpack the contents of the software zip archive. In case you don’t have it you can download it from here: https://github.com/DSD-BookEasy/BookEasy/archive/master.zip
- The archive provides only the parts of software that have been developed by the DSD team and a general skeleton of a Yii project. Additional dependencies, including the actual code of the Yii2 Framework, must be downloaded separately using the dependencies manager of php: *composer*
    - Download composer from here: https://getcomposer.org/download/
    - And put the *composer.phar* file in the same folder where you unpacked the software and make sure it has the permissions to be executed
    - Open a shell and cd to the folder where you unpacked the software
    - Execute the following commands to get all the software dependencies for the product:

      ```
      composer global require "fxp/composer-asset-plugin:1.0.0-beta4"

      composer install
      ```
- The next step is to configure the database. You first need an empty database on your MySql server with a user with usage rights on it
- To configure the database rename the file /config/db.example.php into /config/db.php and open it with a text editor
    - You should complete the lines starting with ‘dsn’, ‘username’ and ‘password’
    - ‘username’ line should contain the name of the database user used to access the database. Substitute the word root with the name of your user
    - ‘password’ line should contain the password of the user that you put in the ‘username’ line. Put it between the two apostrophes
    - ‘dsn’ line should contain the connection string to the database. Except for particular cases it should be structured in this way (capital words should be replaced with your configuration values):

      ```
      mysql:host=HOSTNAME;dbname=DATABASE
      ```
    - Save the file and close it
- Make sure the database server is online and from the same shell you opened before execute the following commands to initialize the database:

  ```
  ./yii migrate --migrationPath=@yii/rbac/migrations/

  ./yii migrate
  ```
- The last step of configuration for Apache is to make sure the *DocumentRoot* configuration of the website is pointing to the absolute path of the *web/* folder of the product

    - If it points to the folder where you unpacked the product it will not work!
    - Also make sure there is a <Directory> section for that folder allowing access to it
    - The Apache configuration for the website should look like this (in this example the product is configured as Virtual Host):

      ```
      <VirtualHost *:80>
              ServerAdmin webmaster@localhost
              ServerName bookeasy.flygmuseum.com

              DocumentRoot /var/www/web
              <Directory /var/www/web>
                      Options Indexes FollowSymLinks MultiViews
                      AllowOverride All
                      Order allow,deny
                      allow from all
              </Directory>

              ErrorLog ${APACHE_LOG_DIR}/error.log

              # Possible values include: debug, info, notice, warn,
      error, crit,
              # alert, emerg.
              LogLevel warn

              CustomLog ${APACHE_LOG_DIR}/access.log combined
      </VirtualHost>
      ```

- Remember to restart Apache after you have changed its configuration
- To ensure your system is secure you should open the *config/web.php* file with a text editor and find the line starting with ‘cookieValidationKey’. Change the sequence of random letters and numbers with a new random sequence and then save the file.
- Now the software should be correctly configured and working. Verify this by opening a browser and typing its address (in the example above it is http://bookeasy.flygmuseum.com)
    - You should be able to see the welcome page of the system
- The system requires to have one Cron Job defined to generate automatically the periodic Timeslots that will be defined for the bookings. You need to create a cron job that makes and http request to the /timeslot-model/generate page of the system (In the example above it would be http://bookeasy.flygmuseum.com/timeslot-model/generate)
    - On linux usually the command to issue in the cron job is something like

      ```
      wget -O - -q http://bookeasy.flygmuseum.com/timeslot-model/generate
      >/dev/null 2>&1
      ```

Additional Configuration

- The system is empty though, with the following steps we will guide you in making the first user with which you will manage the booking system
  - visit the the /setup page (in the example above it would be http://bookeasy.flygmuseum.com/setup)
  - From this page you can create the first user, with its username and password, that will be granted full access to all the functions of the software
- Now you can login with your new user and access the administration section of the system
  - From the System menu above you can access the System Parameters section where you can set some parameters needed by the system to send emails and set the booking calendars
  - From the Permissions entry in the menu you can manage what your future users will be able to do in the system. By default we have created three roles in the system "Admin" (full access, you), "Coordinator" and "Instructor". Simply flag the checkboxes corresponding to the permission you want to grant to each type of user and then confirm clicking onthe submit button. (This operation can be repeated in the future as many times as you want)
  - If you need more roles to manage users and permission in your system you can create/update/delete roles by clicking on the Roles menu entry
  - One of the first things you will want to do is adding the data about the simulators available in the museum. This can be done by clicking on the Simulators menu entry in the System menu
  - You will then want to define when each simulator is available. You can do this by defining
- As you may have noticed the system is all in english. We know you need it in Swedish so we have set it up ready to be translated:
  - In the /messages/sv-SE/messages.po file there are all the sentences and words used in the system, ready to be translated
  - The po file is a standard Gettext file, that is a common format used in GNU projects to support translations. You just need to edit this file to have the whole system translated in Swedish. There exist programs that can help you in editing this file, there are even online services. We highly recommend that you use these programs so that you are sure you don't corrupt the file format by mistake
  - To officially enable to Swedish language in the system, once you have the translated file, you need to open the */config/web.php* file with a text editor and delete the two slashes (//) (only the slashes!!) from the line saying: `'language' => 'sv-SE',`
  - In this way the Swedish sentences will be shown in the website

Additional Notes

The development repository of the software is located here on GitHub:
https://github.com/DSD-BookEasy/BookEasy